IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Adkins et al.**	§	
	§	Group Art Unit: **2163**
Serial No. **10/777,719**	§	
	§	Examiner: **Black, Linh**
Filed: **February 12, 2004**	§	
	§	
For: **Method and Apparatus for File System Snapshot Persistence**	§	

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

35525
PATENT TRADEMARK OFFICE
CUSTOMER NUMBER

APPEAL BRIEF (37 C.F.R. 41.37)

This brief is in furtherance of the Notice of Appeal, filed in this case on February 9, 2007.

A fee of $500.00 is required for filing an Appeal Brief. Please charge this fee to IBM Corporation Deposit Account No. 09-0447. No additional fees are believed to be necessary. If, however, any additional fees are required, I authorize the Commissioner to charge these fees which may be required to IBM Corporation Deposit Account No. 09-0447. No extension of time is believed to be necessary. If, however, an extension of time is required, the extension is requested, and I authorize the Commissioner to charge any fees for this extension to IBM Corporation Deposit Account No. 09-0447.

REAL PARTY IN INTEREST

The real party in interest in this appeal is the following party: International Business Machines Corporation of Armonk, New York.

RELATED APPEALS AND INTERFERENCES

With respect to other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interferences.

STATUS OF CLAIMS

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

Claims in the application are: 1-24.

B. STATUS OF ALL THE CLAIMS IN APPLICATION

1. Claims canceled: None.
2. Claims withdrawn from consideration but not canceled: None.
3. Claims pending: 1-24.
4. Claims allowed: None.
5. Claims rejected: 1-24.
6. Claims objected to: None.

C. CLAIMS ON APPEAL

The Claims on appeal are: 1-24.

STATUS OF AMENDMENTS

No amendments were made after the Final Office Action dated December 29, 2006.

SUMMARY OF CLAIMED SUBJECT MATTER

A. CLAIM 1 - INDEPENDENT

The subject matter of claim 1 is directed to a method in a data processing system (**100**, **200**) for managing data in a file system (**318**). A request is detected to modify a data block in the file system (see *Specification*, page 16, lines 27-29; and **step 900**). Responsive to detecting the request, metadata describing the data block in the file system is written into a snapshot image (**320**, **400**, **500**, **600**, **700**, **800**). The snapshot image is updated to maintain a consistent block-level image of the file system from a point-in-time when the snapshot was created (see *Specification*, page 11, lines 1-23; page 16, line 21, through page 19, line 7; and **Figures 9A** and **9B**). Responsive to detecting the request, data for the data block in the file system is copied to the snapshot image to further update the snapshot image (see *Specification*, page 11, lines 1-23; page 16, line 21, through page 19, line 7; and **Figures 9A** and **9B**). The data block in the file system is modified after copying of the data in the data block to the snapshot image has occurred (see *Specification*, page 11, lines 18-20). The snapshot image is usable to return the file system to a state prior to modifying the data block in the file system (see *Specification*, page 11, lines 23-26).

B. CLAIM 12 - INDEPENDENT

The subject matter of claim 12 is directed to a data processing system (**100**, **200**) for managing data in a file system (**318**). The data processing system provides means for detecting a request to modify a data block in the file system (see *Specification*, page 16, lines 27-29; and **step 900**). Responsive to detecting the request, the data processing system provides means for writing metadata describing the data block in the file system into a snapshot image (**320**, **400**, **500**, **600**, **700**, **800**). The snapshot image is updated to maintain a consistent block-level image of the file system from a point-in-time when the snapshot was created (see *Specification*, page 11, lines 1-23; page 16, line 21, through page 19, line 7; and **Figures 9A** and **9B**). Responsive to detecting the request, the data processing system provides means for copying data for the data block in the file system to the snapshot image to further update the snapshot image (see *Specification*, page 11, lines 1-23; page 16, line 21, through page 19, line 7; and **Figures 9A** and

9B). The data processing system provides means for modifying the data block in the file system after copying of the data in the data block to the snapshot image has occurred (see *Specification*, page 11, lines 18-20). The snapshot image is usable to return the file system to a state prior to modifying the data block in the file system (see *Specification*, page 11, lines 23-26).

C. CLAIM 18 - INDEPENDENT

The subject matter of claim 18 is directed to a computer program product in a computer readable medium for managing data in a file system (**318**) in a data processing system (**100, 200**). The computer program product provides first instructions for detecting a request to modify a data block in the file system (see *Specification*, page 16, lines 27-29; and **step 900**). Responsive to detecting the request, the computer program product provides second instructions for writing metadata describing the data block in the file system into a snapshot image (**320, 400, 500, 600, 700, 800**). The snapshot image is updated to maintain a consistent block-level image of the file system from a point-in-time when the snapshot was created (see *Specification*, page 11, lines 1-23; page 16, line 21, through page 19, line 7; and **Figures 9A** and **9B**). Responsive to detecting the request, the computer program product provides third instructions for copying data for the data block in the file system to the snapshot image to further update the snapshot image (see *Specification*, page 11, lines 1-23; page 16, line 21, through page 19, line 7; and **Figures 9A** and **9B**). The computer program product provides fourth instructions for modifying the data block in the file system after copying of the data in the data block to the snapshot image has occurred (see *Specification*, page 11, lines 18-20). The snapshot image is usable to return the file system to a state prior to modifying the data block in the file system (see *Specification*, page 11, lines 23-26).

D. CLAIM 24 - INDEPENDENT

The subject matter of claim 24 is directed to a data processing system (**100, 200**) comprising: a bus system; a memory connected to the bus system; and a processing unit connected to the bus system. The memory includes a set of instructions. The processing unit executes a set of instructions to detect a request to modify a data block in the file system (**318**) (see *Specification*, page 16, lines 27-29; and **step 900**). The processing unit executes a set of

instructions to write metadata describing the data block in the file system into a snapshot image (**320**, **400**, **500**, **600**, **700**, **800**) in response to detecting the request. The snapshot image is updated to maintain a consistent block-level image of the file system from a point-in-time when the snapshot was created (see *Specification*, page 11, lines 1-23; page 16, line 21, through page 19, line 7; and **Figures 9A** and **9B**). The processing unit executes a set of instructions to copy data for the data block in the file system to the snapshot image to further update the snapshot image in response to detecting the request (see *Specification*, page 11, lines 1-23; page 16, line 21, through page 19, line 7; and **Figures 9A** and **9B**). The processing unit executes a set of instructions to modify the data block in the file system after copying of the data in the data block to the snapshot image has occurred (see *Specification*, page 11, lines 18-20). The snapshot image is usable to return the file system to a state prior to modifying the data block in the file system (see *Specification*, page 11, lines 23-26).

E. CLAIM 3 - DEPENDENT

The subject matter of claim 3, which depends from claim 1, is directed to a method wherein the snapshot image includes a snapshot summary map (**402**, **506**), a snapshot map (**404**), and a set of segments (**406**, **801**) and wherein the summary map identifies initialized states for snapshot map pages (**602**, **604**, **606**, **608**, **700**) in the snapshot map, the snapshot map contains the snapshot map pages that identify data blocks in use in the file system, and the set of segments includes copies of data blocks from the file system (see *Specification*, page 12, line 9, through page 16, line 20; and **Figures 4**, **5**, **6**, **7** and **8**).

F. CLAIM 7 - DEPENDENT

The subject matter of claim 7, which depends from claim 1, is directed to a method wherein the writing step comprises: writing an in-use state of snapshot map entries (**702**, **704**, **706**, **708**, **710**, **712**, **714**, **716**, **718**, **720**, **722**, **724**) for a snapshot map group (**600**) to the snapshot image prior to any before-image data blocks referenced by the snapshot map group being written to the snapshot image (see *Specification*, page 12, lines 18-22; page 13, line 27, through page 15, line 6; and **Figures 6** and **7**).

G. CLAIM 8 - DEPENDENT

The subject matter of claim 8, which depends from claim 1 through claim 7, is directed to a method wherein the writing step further comprises: marking a summary snapshot map entry (**702**, **704**, **706**, **708**, **710**, **712**, **714**, **716**, **718**, **720**, **722**, **724**) as being initialized and marking a location (**726**, **728**, **730**) of the snapshot map group after writing the in-use state of data blocks for the snapshot map group to the snapshot image (see *Specification*, page 12, lines 18-22; page 13, line 27, through page 15, line 6; and **Figures 6** and **7**).

H. CLAIM 14 - DEPENDENT

The subject matter of claim 14, which depends from claim 12, is directed to a data processing system wherein the snapshot image includes a snapshot summary map (**402**, **506**), a snapshot map (**404**), and a set of segments (**406**, **801**) and wherein the summary map identifies initialized states for snapshot map pages (**602**, **604**, **606**, **608**, **700**) in the snapshot map, the snapshot map contains the snapshot map pages that identify data blocks in use in the file system, and the set of segments includes copies of data blocks from the file system (see *Specification*, page 12, line 9, through page 16, line 20; and **Figures 4**, **5**, **6**, **7** and **8**).

I. CLAIM 20 - DEPENDENT

The subject matter of claim 20, which depends from claim 18, is directed to a computer program product wherein the snapshot image includes a snapshot summary map (**402**, **506**), a snapshot map (**404**), and a set of segments (**406**, **801**), wherein the summary map identifies initialized states for snapshot map pages (**602**, **604**, **606**, **608**, **700**) in the snapshot map, the snapshot map contains the snapshot map pages that identify data blocks in use in the file system or data blocks copied into the snapshot, and the set of segments includes copies of data blocks from the file system (see *Specification*, page 12, line 9, through page 16, line 20; and **Figures 4**, **5**, **6**, **7** and **8**).

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The grounds of rejection to review on appeal are as follows:

A. GROUND OF REJECTION 1 (Claims 1-2, 7, 12-13, 18-19 and 24)

Claims 1-2, 7, 12-13, 18-19 and 24 stand rejected under 35 U.S.C. § 102(e) as being anticipated over *McGovern et al.*, System and Method for Record Retention Date in a Write Once Read Many Storage System, U.S. Patent Publication No. 2005/0097260 A1 (published, May 5, 2005).

B. GROUND OF REJECTION 2 (Claims 3-6, 8-11, 14-17 and 20-23)

Claims 3-6, 8-11, 14-17 and 20-23 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over *McGovern et al.* (US 2005/0097260), and further in view of *Edwards*, System and Method for Performing an On-line Check of a File System, U.S. Patent Publication No. 2003/0182389 A1 (published, September 25, 2003).

ARGUMENT

A. GROUND OF REJECTION 1 (1-2, 7, 12-13, 18-19, and 24)

The Final Office Action rejects claims 1-2, 7, 12-13, 18-19, and 24 under 35 U.S.C. §102(e) as being anticipated by *McGovern et al.* (US 2005/0097260), hereinafter referred to as *McGovern*. This rejection is respectfully traversed.

A.1. Claims 1-2, 12-13, 18-19, and 24

With respect to independent claims 1, 12, 18, and 24, the Final Office Action states:

> **As per claim 1**, McGovern et al. teach detecting a request to modify a data block in the file system; responsive to detecting the request – pars. 0006, 0045-0046, 0081, 0054 (block-based files); writing metadata describing the data block in the file system into a snapshot image – pars. 0020 (writing metadata), 0050, 0058-0063 (snapshot, a persistent consistency point image; PCPI: a persistent consistency point image is a point in time representation of the storage system and particularly, of the active file system stored on a storage device…PCPI can also include other information: metadata about the active file system at the particular point in time for which the image is taken), 0068, 0078, 0081 (request for modifications), 0104, 0122 (the metadata file attributes or properties to determine the last modified time on the file), 0115 (metadata).
>
> wherein the snapshot image is updated to maintain a consistent block level image of the file system from a point in time when the snapshot was created – pars. 0058-0063.
>
> copying data for the data block in the file system to the snapshot image to further update the snapshot image – pars. 0046, 0058-0063 (one common form of update involves the use of a snapshot process in which the active file system at the storage site, consisting of inodes and blocks…), 0062-0063.
>
> and modifying the data block in the file system after copying of the data in the data block to the snapshot image has occurred, wherein the snapshot image is usable to return the file system to a state prior to modifying the data block in the file system – pars. 0058-0063 (snapshot, a persistent consistency point image; PCPI: a persistent consistency point image is a point in time representation of the storage system and particularly, of the active file system stored on a storage device…PCPI can also include other information: metadata about the active file system at the particular point in time for which the image is taken), 0068, **0078**,

0081 (request for modifications), 0104, 0122 (the metadata file attributes or properties to determine the last modified time on the file), 0078, 0114.
...

Claims 12-24 are rejected based on the same rationale as claims 1-9.

Final Office Action dated December 29, 2006, pages 2-3 and page 7.

Claim 1, which is representative of the other rejected independent claims 12, 18, and 24 with regard to similarly recited subject matter, reads as follows:

> 1. A method in a data processing system for managing data in a file system, the method comprising:
> detecting a request to modify a data block in the file system;
> responsive to detecting the request:
> writing metadata describing the data block in the file system into a snapshot image, wherein the snapshot image is updated to maintain a consistent block-level image of the file system from a point-in-time when the snapshot was created; and
> copying data for the data block in the file system to the snapshot image to further update the snapshot image; and
> modifying the data block in the file system after copying of the data in the data block to the snapshot image has occurred, wherein the snapshot image is usable to return the file system to a state prior to modifying the data block in the file system.
> (emphasis added)

A prior art reference anticipates the claimed invention under 35 U.S.C. § 102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Anticipation focuses on whether a claim reads on the product or process a prior art reference discloses, not on what the reference broadly teaches. *Kalman v. Kimberly-Clark Corp.*, 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983). Appellants respectfully submit that *McGovern* does not identically show every element of the claimed invention arranged as they are in the claims. Specifically, *McGovern* does not teach or suggest "responsive to detecting the request: writing metadata describing the data block in the file system into a snapshot image, wherein the snapshot image is updated to maintain a

consistent block-level image of the file system from a point-in-time when the snapshot was created; and copying data for the data block in the file system to the snapshot image to further update the snapshot image," as recited in claims 1, 12, 18, and 24.

McGovern is directed to providing a specified retention date within a data set that is locked against deletion or modification within a write-once-read-many (WORM) storage implementation. This retention date scheme does not utilize any proprietary application program interfaces (APIs) or protocols, but rather, employs native functionality within conventional file (or other data containers, data sets or block-based logical unit numbers) properties available in commonly used operating systems. In an illustrative embodiment, the retention date/time is calculated by querying the file's last-modified time prior to commit, adding the retention period to this value and thereby deriving a retention date after which the file can be released from WORM. Prior to commit, the computed retention date is stored in the file's "last access time" property/attribute field, or another metadata field that remains permanently associated with the file and that, in being used for retention date, does not interfere with file management in a WORM state. Since this field is not utilized in a WORM context, it can be adapted to store this date. Once stored, the retention date in this field is locked against modification. Where extension (never reduction) of a retention period is desired, the last access time field is updated, wherein the new retention period is added to the existing last access time value to derive a new, later retention date for the file. Upon expiry of the retention date, the system allows deletion of the expired WORM file/data set. *McGovern* mentions that a snapshot is an image of a file system at a point in time. *McGovern* discloses that each time a snapshot is taken, the old active file system becomes the new snapshot. *McGovern* does not teach or suggest that a snapshot image is updated in response to detecting a request to modify a data block in the file system so that the snapshot maintains a consistent block-level image of the file system. In addition, *McGovern* does not teach or suggest further updating the snapshot image in response to the request to modify the data block by copying data for the data block to the snapshot image. In the claimed invention, these steps are performed prior to completing the request to modify the data block. The data block is modified after the copying of the data to the snapshot image to further update the snapshot image. *McGovern* does not teach or suggest that responsive to detecting the request (to modify a data block in the file system): writing metadata describing the data block in

the file system into a snapshot image, wherein the snapshot image is updated to maintain a consistent block-level image of the file system from a point-in-time when the snapshot was created; and copying data for the data block in the file system to the snapshot image to further update the snapshot image, as recited in claims 1, 12, 18, and 24.

The Final Office Action refers to the following portions of *McGovern* in the rejection of independent claims 1, 12, 18, and 24:

> [0058] One common form of update involves the use of a "snapshot" process in which the active file system at the storage site, consisting of inodes and blocks, is captured and the "snapshot" is transmitted as a whole, over a network (such as the well-known Internet) to the remote storage site. Generally, a snapshot is an image (typically read-only) of a file system at a point in time, which is stored on the same primary storage device as is the active file system and is accessible by users of the active file system. By "active file system" it is meant the file system to which current input/output operations are being directed. The primary storage device, e.g., a set of disks, stores the active file system, while a secondary storage, e.g., a tape drive, may be utilized to store backups of the active file system. Once the snapshot is taken (i.e., the image captured), the active file system is reestablished, leaving the snapshotted version in place for possible disaster recovery. Each time a snapshot is taken, the old active file system becomes the new snapshot, and the new active file system carries on, recording any new changes. A set number of snapshots may be retained depending upon various time-based and other criteria. (emphasis added)
>
> [0059] "Snapshot" is a trademark of Network Appliance, Inc. It is used for purposes of this patent to designate a persistent consistency point (CP) image. A persistent consistency point image (PCPI) is a point-in-time representation of the storage system, and more particularly, of the active file system, stored on a storage device (e.g., on disk) or in other persistent memory and having a name or other unique identifier that distinguishes it from other PCPIs taken at other points in time. A PCPI can also include other information (metadata) about the active file system at the particular point in time for which the image is taken. The terms "PCPI" and "snapshot" shall be used interchangeably through out this patent without derogation of Network Appliance's trademark rights.

McGovern, paragraphs [0058] – [0059].

These portions of *McGovern* disclose that a snapshot is an image of a file system at a point in time, which may be used for disaster recovery. A file system may be updated for disaster recovery using a snapshot. Each time a snapshot is taken, the old active file system

becomes the new snapshot. In other words, *McGovern* teaches replacing a previous snapshot with a new snapshot. *McGovern* does not teach or suggest updating a snapshot image or updating a snapshot image in response to detecting a request to modify a data block in the file system. *McGovern* does not teach or suggest that responsive to detecting the request (to modify the data block in the file system): writing metadata describing the data block in the file system into a snapshot image, wherein the snapshot image is updated to maintain a consistent block-level image of the file system from a point-in-time when the snapshot was created; and copying data for the data block in the file system to the snapshot image to further update the snapshot image, as recited in claims 1, 12, 18, and 24.

In addition, the Final Office Action refers to the following portions of *McGovern* in the rejection of independent claims 1, 12, 18, and 24:

> [0060] A PCPI/snapshotting process is described in further detail in U.S. patent application Ser. No. 09/932,578, entitled INSTANT SNAPSHOT by Blake Lewis et al., which is hereby incorporated by reference as though fully set forth herein. In addition, the native Snapshot™ capabilities of the WAFL file system are further described in TR3002 File System Design for an NFS File Server Appliance by David Hitz et al., published by Network Appliance, Inc., and in commonly owned U.S. Pat. No. 5,819,292, entitled METHOD FOR MAINTAINING CONSISTENT STATES OF A FILE SYSTEM AND FOR CREATING USER-ACCESSIBLE READ-ONLY COPIES OF A FILE SYSTEM by David Hitz et al., which are hereby incorporated by reference.
>
> [0061] With renewed reference to FIG. 2, overlying the file system layer 250 is the user interface 285 for the administrator. This can be accessed via the various protocols (CIFS, NFS, etc.) described above.
>
> [0062] Also, overlying the file system layer 250 is a specialized asynchronous volume and sub-volume (or "qtree") snapshot mirroring (or replication) application 290. This application is responsible for the generation of the mirrors at a remote destination storage volume based upon changes in snapshots at the source. The snapshot mirroring application 290 operates outside of the storage access request path 270, as shown by the direct links 292 and 294 to the TCP/IP layers 215, 210 and the file system snapshot mechanism (280). This application enables "asynchronous" mirroring of changes in the respective sub-volume. That is mirroring is written incrementally (and not in real-time with respect to changes occurring on the source sub-volume) to a destination store that can be remote and linked by a network connection. A discussion of asynchronous mirroring at a volume and sub-volume (or q-tree) level is

> found in U.S. patent application Ser. No. 10/100,967, entitled SYSTEM AND METHOD FOR DETERMINING CHANGES IN TWO SNAPSHOTS AND FOR TRANSMITTING CHANGES TO A DESTINATION SNAPSHOT by Michael L. Federwisch, et al., which is hereby incorporated by reference.
>
> [0063] Likewise a synchronous volume snapshot mirroring application 298 acting on the RAID layer is provided. This application provides synchronous (real-time) mirroring to a mirror store in response to a mirror command initiated by an administrator. This mirroring creates a point-in-time image of the source that copies it as it existed at the time the mirror command is acted upon. Approaches to volume-based remote mirroring of snapshots are described in detail in commonly owned U.S. patent application Ser. No. 09/127,497, entitled FILE SYSTEM IMAGE TRANSFER by Steven Kleiman, et al. and U.S. patent application Ser. No. 09/426,409, entitled FILE SYSTEM IMAGE TRANSFER BETWEEN DISSIMILAR FILE SYSTEMS by Steven Kleiman, et al., both of which patents are expressly incorporated herein by reference.

McGovern, paragraphs [0060] – [0063].

These portions of *McGovern* disclose that snapshot asynchronous mirroring is not written in real-time with respect to changes occurring on the source sub-volume. Snapshot synchronous (real-time) mirroring occurs in response to a mirror command initiated by an administrator. *McGovern* discloses that mirroring creates a point-in-time image of the source that copies it as it existed at the time the mirror command is acted upon. *McGovern* does not teach or suggest that responsive to detecting the request (to modify a data block in the file system): writing metadata describing the data block in the file system into a snapshot image, wherein the snapshot image is updated to maintain a consistent block-level image of the file system from a point-in-time when the snapshot was created; and copying data for the data block in the file system to the snapshot image to further update the snapshot image, as recited in claims 1, 12, 18, and 24.

Other portions of *McGovern* discuss a write-once-read-many (WORM) volume and files. These portions are not directed to a snapshot image. Further, *McGovern* states that mirroring and backup functions that enable restoration or reversion of the WORM volume or files from a snapshot to an earlier point in time may be disabled. See *McGovern,* paragraphs [0014] and [0078].

In view of the above, Appellants respectfully submit that *McGovern* does not teach each and every feature of independent claims 1, 12, 18, and 24, as is required under 35 U.S.C § 102(e). In addition, *McGovern* does not teach each and every feature of dependent claims 2, 7, 13, and 19 at least by virtue of their dependency on claims 1, 12, 18, and 24, respectively. Accordingly, Appellants respectfully request withdrawal of the rejection of claims 1-2, 7, 12-13, 18-19, and 24 under 35 U.S.C § 102(e).

Furthermore, *McGovern* does not teach, suggest, or give any incentive to make the needed changes to reach the presently claimed invention. *McGovern* actually teaches away from the presently claimed invention because it teaches that a new snapshot image records any new changes to a file system and replaces a previous snapshot image opposed to continuously updating a single snapshot image to maintain a persistent snapshot and a consistent block-level image of a file system as in the presently claimed invention. Absent the examiner pointing out some teaching or incentive to implement *McGovern* and continuously updating a snapshot image to maintain a persistent snapshot and a consistent block-level image of a file system, one of ordinary skill in the art would not be led to modify *McGovern* to reach the present invention when the reference is examined as a whole. Absent some teaching, suggestion, or incentive to modify *McGovern* in this manner, the presently claimed invention can be reached only through an improper use of hindsight using the Appellants' disclosure as a template to make the necessary changes to reach the claimed invention.

A.2. Claim 7

In addition to being dependent on independent claim 1, claim 7 is also distinguished over the *McGovern* reference based on the specific features recited therein. *McGovern* does not teach or suggest that "the writing step comprises: writing an in-use state of snapshot map entries for a snapshot map group to the snapshot image prior to any before-image data blocks referenced by the snapshot map group being written to the snapshot image," as recited in claim 7. To the contrary, *McGovern* does not even mention a snapshot map or writing a state of snapshot map entry for a snapshot map group to the snapshot image. Figure 7 of the present invention, displayed below, depicts a snapshot map page **700** in a snapshot map group **600** that shows example entries with the in-use state. Entries **702**, **704**, **706**, **708**, **710**, **712**, and **714** indicate that

file system blocks 0, 1, 2, 508, 509, 510, and 511 were in-use in the file system when the snapshot was created. Entries within the snapshot map pages describe the in-use and copied state for every data block in the file system.

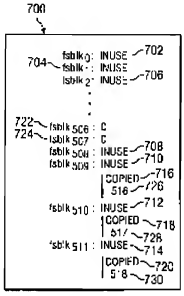

FIG. 7

Specification, Figure 7.

Cited portions of *McGovern* discuss a write-once-read-many state for a file/data set and a retention date within a data set. Any WORM file modifications are restricted. WORM disables conventional restoration of WORM volume or parts thereof from a snapshot. Upon expiry of the retention date, the system allows deletion of the expired WORM file/data set. *McGovern* does not teach or suggest the features of claim 7.

B. GROUND OF REJECTION 2 (Claims 3-6, 8-11, 14-17, and 20-23)

The Final Office Action rejects claims 3-6, 8-11, 14-17, and 20-23 under 35 U.S.C. §103(a) as being unpatentable over *McGovern* and further in view of *Edwards* (US 2003/0182389). This rejection is respectfully traversed.

B.1. Claims 5-6, 7-11, 15-17, and 21-23

Since claims 3-6, 8-11, 14-17, and 20-23 depend from independent claims 1, 12, and 18, respectively, the same distinctions between *McGovern* and the invention recited in claims 1, 12,

and 18 apply to dependent claims 3-6, 8-11, 14-17, and 20-23. In addition, *Edwards* does not provide for the deficiencies of *McGovern* with regard to independent claims 1, 12, and 18. *Edwards* is directed to a system and method for performing an on-line check of a file system. Various function calls are modified within a file system layer of a storage operating system so that each time the particular inode is retrieved using the modified function calls, a check is performed on the inode and associated buffer trees before returning the requested inode to the calling process. *Edwards* is cited for disclosing a summary map and a snap map. *Edwards* does not teach or suggest "responsive to detecting the request: writing metadata describing the data block in the file system into a snapshot image, wherein the snapshot image is updated to maintain a consistent block-level image of the file system from a point-in-time when the snapshot was created; and copying data for the data block in the file system to the snapshot image to further update the snapshot image," as recited in claims 1, 12, and 18. Thus, any alleged combination of *McGovern* with *Edwards* still would not result in the invention recited in claims 1, 12, and 18 from which claims 3-6, 8-11, 14-17, and 20-23 depend. Accordingly, Appellants respectfully request withdrawal of the rejection of claims 3-6, 8-11, 14-17, and 20-23 under 35 U.S.C. § 103(a).

B.2. Claims 3, 8, 14, and 20

In addition to being dependent on their respective independent claims, claims 3, 8, 14, and 20 are also distinguished over the *McGovern* and *Edwards* references based on the specific features recited therein. Claims 3 and 8 are dependent on independent claim 1; claim 14 is dependent on independent claim 12; and claim 20 is dependent on independent claim 18. *McGovern* and *Edwards*, taken alone or in combination, do not teach or suggest that the snapshot image includes a snapshot summary map, a snapshot map, and a set of segments and that the snapshot summary map identifies initialized states for snapshot map pages in the snapshot map, the snapshot map contains the snapshot map pages that identify data blocks in use in the file system, and the set of segments includes copies of data blocks from the file system, as recited in claims 3, 14, and 20. Similarly, *McGovern* and *Edwards*, taken alone or in combination, do not teach or suggest marking a summary snapshot map entry as being initialized and marking a location of the snapshot map group after writing the in-use state of data blocks for the snapshot

map group to the snapshot image, as recited in claim 8. To the contrary, *McGovern* and *Edwards* do not mention initialized states for snapshot map pages in the snapshot map. The Final Office Action states that *McGovern* does not teach a summary map or a snapshot map. *Edwards* discloses that a file system is checked for errors in response to user input, a system crash, or other error condition and that an inode is marked as being checked. This marking can be accomplished by modifying a tracking file or by modifying a bit within the inode's metadata. *Edwards* discloses that the summary map stores metadata associated with determining which blocks are used by snapshots. *McGovern* and *Edwards*, taken alone or in combination, do not teach or suggest the specific features of claims 3, 8, 14, and 20.

/Gerald H. Glanzman/
Gerald H. Glanzman
Reg. No. 25,035
YEE & ASSOCIATES, P.C.
PO Box 802333
Dallas, TX 75380
(972) 385-8777

GG/VA

CLAIMS APPENDIX

The text of the claims involved in the appeal are:

1. A method in a data processing system for managing data in a file system, the method comprising:

detecting a request to modify a data block in the file system;

responsive to detecting the request:

writing metadata describing the data block in the file system into a snapshot image, wherein the snapshot image is updated to maintain a consistent block-level image of the file system from a point-in-time when the snapshot was created; and

copying data for the data block in the file system to the snapshot image to further update the snapshot image; and

modifying the data block in the file system after copying of the data in the data block to the snapshot image has occurred, wherein the snapshot image is usable to return the file system to a state prior to modifying the data block in the file system.

2. The method of claim 1, wherein the copying step includes:

storing an identification of information used to locate the data in a table within the snapshot image, wherein the table is used to locate different data blocks in the file system.

3. The method of claim 1, wherein the snapshot image includes a snapshot summary map, a snapshot map, and a set of segments and wherein the summary map identifies initialized states

for snapshot map pages in the snapshot map, the snapshot map contains the snapshot map pages that identify data blocks in use in the file system, and the set of segments includes copies of data blocks from the file system.

4. The method of claim 1, wherein the metadata includes a snapshot summary map, a snapshot map, and segment headers and wherein the metadata of the snapshot image is used to reconstruct the state of the file system in response to a selected event.

5. The method of claim 4, wherein the selected event is a failure of the data processing system while modifying the data block in the file system.

6. The method of claim 4, wherein the selected event is a user input requesting restoration of the file system to a consistent state.

7. The method of claim 1, wherein the writing step comprises:

writing an in-use state of snapshot map entries for a snapshot map group to the snapshot image prior to any before-image data blocks referenced by the snapshot map group being written to the snapshot image.

8. The method of claim 7, wherein the writing step further comprises:

marking a summary snapshot map entry as being initialized and marking a location of the snapshot map group after writing the in-use state of data blocks for the snapshot map group to the snapshot image.

9. The method of claim 8, wherein the writing step further comprises:

initializing a segment header of a new last segment in a list of segments before a prior segment is modified to point to the new last segment.

10. The method of claim 4, wherein the snapshot map contains snapshot map pages and wherein the snapshot map pages are reconstructed during a recovery operation.

11. The method of claim 10, wherein the recovery operation handles copying of before-images of data blocks in the file system that are to be modified by the recovery operation.

12. A data processing system for managing data in a file system, the data processing system comprising:

detecting means for detecting a request to modify a data block in the file system;

responsive to detecting the request:

writing means for writing metadata describing the data block in the file system into a snapshot image, wherein the snapshot image is updated to maintain a consistent block-level image of the file system from a point-in-time when the snapshot was created; and

copying means for copying data for the data block in the file system to the snapshot image to further update the snapshot image; and

modifying means for modifying the data block in the file system after copying of the data in the data block to the snapshot image has occurred, wherein the snapshot image is usable to return the file system to a state prior to modifying the data block in the file system.

13. The data processing system of claim 12, wherein the copying means includes:

storing means for storing an identification of information used to locate the data in a table within the snapshot image, wherein the table is used to locate different data blocks in the file system.

14. The data processing system of claim 12, wherein the snapshot image includes a snapshot summary map, a snapshot map, and a set of segments and wherein the summary map identifies initialized states for snapshot map pages in the snapshot map, the snapshot map contains the snapshot map pages that identify data blocks in use in the file system, and the set of segments includes copies of data blocks from the file system.

15. The data processing system of claim 12, wherein the metadata includes a snapshot summary map, a snapshot map, and segment headers and wherein the metadata of the snapshot image is used to reconstruct the state of the file system in response to a selected event.

16. The data processing system of claim 15, wherein the selected event is a failure of the data processing system while modifying the data block in the file system.

17. The data processing system of claim 15, wherein the selected event is a user input requesting restoration of the file system to a consistent state.

18. A computer program product in a computer readable medium for managing data in a file system in a data processing system, the computer program product comprising:

first instructions for detecting a request to modify a data block in the file system;

responsive to detecting the request:

second instructions for writing metadata describing the data block in the file system into a snapshot image, wherein the snapshot image is updated to maintain a consistent block-level image of the file system from a point-in-time when the snapshot was created; and

third instructions for copying data for the data block in the file system to the snapshot image to further update the snapshot image; and

fourth instructions for modifying the data block in the file system after copying of the data in the data block to the snapshot image has occurred, wherein the snapshot image is usable to return the file system to a state prior to modifying the data block in the file system.

19. The computer program product of claim 18, wherein the third instructions includes:

sub-instructions for storing an identification of information used to locate the data in a table within the snapshot image, wherein the table is used to locate different data blocks in the file system.

20. The computer program product of claim 18, wherein the snapshot image includes a snapshot summary map, a snapshot map, and a set of segments, wherein the summary map identifies initialized states for snapshot map pages in the snapshot map, the snapshot map contains the snapshot map pages that identify data blocks in use in the file system or data blocks copied into the snapshot, and the set of segments includes copies of data blocks from the file system.

21. The computer program product of claim 18, wherein the metadata includes a snapshot summary map, a snapshot map, and segment headers and wherein the metadata of the snapshot image is used to reconstruct the state of the file system in response to a selected event.

22. The computer program product of claim 21, wherein the selected event is a failure of the data processing system while modifying the data block in the file system.

23. The computer program product of claim 21, wherein the selected event is a user input requesting restoration of the file system to a consistent state.

24. A data processing system comprising:

a bus system;

a memory connected to the bus system, wherein the memory includes a set of instructions; and

a processing unit connected to the bus system, wherein the processing unit executes a set of instructions to detect a request to modify a data block in the file system; write metadata

describing the data block in the file system into a snapshot image, in response to detecting the request, wherein the snapshot image is updated to maintain a consistent block-level image of the file system from a point-in-time when the snapshot was created; copy data for the data block in the file system to the snapshot image to further update the snapshot image in response to detecting the request; and modify the data block in the file system after copying of the data in the data block to the snapshot image has occurred, wherein the snapshot image is usable to return the file system to a state prior to modifying the data block in the file system.

EVIDENCE APPENDIX

There is no evidence to be presented.

RELATED PROCEEDINGS APPENDIX

There are no related proceedings.